APPENDICIES

Appendix 1: Abbreviation Key for Mass-Volume Model

Abbreviation
Kf sd = associated rate constant for stomach and duodenum
Ka dj = associated rate constant for duodenum and jejunum
Ka ji = associated rate constant for jejunum and ileum
Ka ie = associated rate constant for ileum and colon
Ka co = associated rate constant for colon and excretion
SD trans = transfer rate between stomach and duodenum
DJ trans = transfer rate between duodenum and jejunum
JL trans = transfer rate between jejunum and ileum
IC trans = transfer rate between ileum and colon
Waste = transfer rate between colon and excretion
pH s = pH stomach
pH s2 = pH duodenum
pH s3 = pH jejunum
pH s4 = pH ileum
pH s5 = pH colon
sol profile = solubility profile for stomach

sol profile 2 = solubility profile for duodenum
sol profile 3 = solubility profile for jejunum
sol profile 4 = solubility profile for ileum
sol profile 5 = solubility profile for colon
stom ka = associated rate constant for stomach compartments 1 and 2
duo ka = associated rate constant for duodenum compartments 1 and 2
Jej ka = associated rate constant for jejunum compartments 1 and 2
Il ka = associated rate constant for ileum compartments 1 and 2
Colon ka = associated rate constant for colon compartments 1 and 2
SA stom = surface area of stomach
SA duo = surface area of duodenum
SA jej = surface area of jejunum
SA il = surface area of ileum
SA colon = surface area of colon
Perm stom = permeability of stomach
Perm duo = permeability of duodenum
Perm jej = permeability of jejunum
Perm il = permeability of ileum
Perm colon = permeability of colon

Ka sd = associated rate construct for stomach fluid absorption
Ka du = associated rate construct for duodeunm fluid absorption
Ka je = associated rate construct for jejunm fluid absorption
Ka il = associated rate construct for ileunm fluid absorption
Ka co = associated rate construct for colon fluid absorption
Note: other abbreviations adhere to above descriptors and are self explanatory

Appendix 2: Equations, Parameters and Values For Mass-Volume Model

amt_plasma(t) = amt_plasma(t - dt) + (trans_21 + ka - elimination - trans_12) * dt
INIT amt_plasma = 0

INFLOWS:
trans_21 = k21*comp_2
ka = tot_abs_rate

OUTFLOWS:
elimination = amt_plasma*k_elim
trans_12 = k12*amt_plasma
blood_side_col(t) = blood_side_col(t - dt) + (colon_ka_5) * dt
INIT blood_side_col = 0

INFLOWS:
colon_ka_5 = IF Vol_colon*sol_profile_5 >=Colon THEN Colon*SA_colon*perm_colon*3600 ELSE Vol_colon*sol_profile_5*SA_colon*perm_colon*3600
blood_side_dou(t) = blood_side_dou(t - dt) + (duo_ka) * dt
INIT blood_side_dou = 0

INFLOWS:
duo_ka = IF Vol_duod*sol_profile_2 >= duodenum THEN duodenum*SA_duo*perm_duo*3600 ELSE Vol_duod*sol_profile_2*SA_duo*perm_duo*3600
blood_side_il(t) = blood_side_il(t - dt) + (Il_ka) * dt
INIT blood_side_il = 0

INFLOWS:
Il_ka = IF Vol_ileum*sol_profile_4 >=Ileum THEN Ileum*SA_Il*perm_Il*3600 ELSE Vol_ileum*sol_profile_4*SA_Il*perm_Il*3600
blood_side_jej(t) = blood_side_jej(t - dt) + (Jej_ka) * dt
INIT blood_side_jej = 0

INFLOWS:
Jej_ka = IF Vol_jej*sol_profile_3 >=Jejunum THEN Jejunum*SA_jej*perm_jej *3600 ELSE Vol_jej*sol_profile_3*SA_jej*perm_jej*3600
blood_side_sto(t) = blood_side_sto(t - dt) + (stom_ka) * dt
INIT blood_side_sto = 0

INFLOWS:
stom_ka = IF Vol_stom*sol_profile >= Stomach THEN Stomach*SA_stom*perm_stom*3600 ELSE Vol_stom*sol_profile*SA_stom*perm_stom*3600
Colon(t) = Colon(t - dt) + (IC_trans - Waste - colon_ka_5) * dt
INIT Colon = 0

```
INFLOWS:
IC_trans = ka_ic*Ileum

OUTFLOWS:
Waste = ka_col*Colon
colon_ka_5 = IF Vol_colon*sol_profile_5 >=Colon THEN Colon*SA_colon*perm_colon*3600
ELSE Vol_colon*sol_profile_5*SA_colon*perm_colon*3600
comp_2(t) = comp_2(t - dt) + (trans_12 - trans_21) * dt
INIT comp_2 = 0

INFLOWS:
trans_12 = k12*amt_plasma

OUTFLOWS:
trans_21 = k21*comp_2
duodenum(t) = duodenum(t - dt) + (SD_trans - duo_ka - DJ_trans) * dt
INIT duodenum = 0

INFLOWS:
SD_trans = if Stomach >0 then kf_sd*Stomach else 0

OUTFLOWS:
duo_ka = IF Vol_duod*sol_profile_2 >= duodenum THEN
duodenum*SA_duo*perm_duo*3600 ELSE Vol_duod*sol_profile_2*SA_duo*perm_duo*3600
DJ_trans = ka_dj*duodenum
excretion(t) = excretion(t - dt) + (vol_cw) * dt
INIT excretion = 0

INFLOWS:
vol_cw = Vol_colon*ka_col
excretion_2(t) = excretion_2(t - dt) + (Waste) * dt
INIT excretion_2 = 0

INFLOWS:
Waste = ka_col*Colon
Ileum(t) = Ileum(t - dt) + (JL_trans - IC_trans - Il_ka) * dt
INIT Ileum = 0

INFLOWS:
JL_trans = ka_ji*Jejunum

OUTFLOWS:
IC_trans = ka_ic*Ileum
Il_ka = IF Vol_ileum*sol_profile_4 >=Ileum THEN Ileum*SA_Il*perm_Il*3600 ELSE
Vol_ileum*sol_profile_4*SA_Il*perm_Il*3600
Jejunum(t) = Jejunum(t - dt) + (DJ_trans - JL_trans - Jej_ka) * dt
```

```
INIT Jejunum = 0

INFLOWS:
DJ_trans = ka_dj*duodenum

OUTFLOWS:
JL_trans = ka_ji*Jejunum
Jej_ka = IF Vol_jej*sol_profile_3 >=Jejunum THEN Jejunum*SA_jej*perm_jej *3600 ELSE
Vol_jej*sol_profile_3*SA_jej*perm_jej*3600
serosal_col(t) = serosal_col(t - dt) + (Adsorp_col - col_secretion) * dt
INIT serosal_col = 0

INFLOWS:
Adsorp_col = PULSE(1.67,0,.1)+0*Vol_colon*ka_co

OUTFLOWS:
col_secretion = 0
serosal_dou(t) = serosal_dou(t - dt) + (Adsorp_Duo - duo_secretion) * dt
INIT serosal_dou = 0

INFLOWS:
Adsorp_Duo = PULSE(10.82,0,.1)+0*Vol_duod*ka_du

OUTFLOWS:
duo_secretion = PULSE(10.82,0,.1)
serosal_ill(t) = serosal_ill(t - dt) + (Adsorpt_ill - ile_secretion) * dt
INIT serosal_ill = 0

INFLOWS:
Adsorpt_ill = PULSE(8.83,0,.10)+0*Vol_ileum*ka_il

OUTFLOWS:
ile_secretion = PULSE(1.50,0,.1)
serosal_jej(t) = serosal_jej(t - dt) + (Adsorp_jej - jej_secretion) * dt
INIT serosal_jej = 0

INFLOWS:
Adsorp_jej = PULSE(15.76,0,.1)+0*Vol_jej*ka_je

OUTFLOWS:
jej_secretion = PULSE(2.67,0,.1)
serosal_sto(t) = serosal_sto(t - dt) + (Adsorp_Stom - Stom_Secretion) * dt
INIT serosal_sto = 0

INFLOWS:
Adsorp_Stom = 0*Vol_stom*ka_sd
```

```
OUTFLOWS:
Stom_Secretion = PULSE(16.67,0,.1)
Stomach(t) = Stomach(t - dt) + (- SD_trans - stom_ka) * dt
INIT Stomach = 1000

OUTFLOWS:
SD_trans = if Stomach >0 then kf_sd*Stomach else 0
stom_ka = IF Vol_stom*sol_profile >= Stomach THEN Stomach*SA_stom*perm_stom*3600
ELSE Vol_stom*sol_profile*SA_stom*perm_stom*3600
total_drug_absorbed(t) = total_drug_absorbed(t - dt) + (tot_abs_rate) * dt
INIT total_drug_absorbed = 0

INFLOWS:
tot_abs_rate = stom_ka+duo_ka+Jej_ka+Il_ka+colon_ka_5
Total_Elimination(t) = Total_Elimination(t - dt) + (elimination) * dt
INIT Total_Elimination = 0

INFLOWS:
elimination =  amt_plasma*k_elim
Vol_colon(t) = Vol_colon(t - dt) + (vol_ij + col_secretion - vol_cw - Adsorp_col) * dt
INIT Vol_colon = 0

INFLOWS:
vol_ij = Vol_ileum*ka_ic
col_secretion = 0

OUTFLOWS:
vol_cw = Vol_colon*ka_col
Adsorp_col = PULSE(1.67,0,.1)+0*Vol_colon*ka_co
Vol_duod(t) = Vol_duod(t - dt) + (vol_sd + duo_secretion - voil_dj - Adsorp_Duo) * dt
INIT Vol_duod = 0

INFLOWS:
vol_sd = kf_sd*Vol_stom
duo_secretion = PULSE(10.82,0,.1)

OUTFLOWS:
voil_dj = Vol_duod*ka_dj
Adsorp_Duo = PULSE(10.82,0,.1)+0*Vol_duod*ka_du
Vol_ileum(t) = Vol_ileum(t - dt) + (vol_ji + ile_secretion - Adsorpt_ill - vol_ij) * dt
INIT Vol_ileum = 0

INFLOWS:
vol_ji = Vol_jej*ka_ji
ile_secretion = PULSE(1.50,0,.1)
```

```
OUTFLOWS:
Adsorpt_ill = PULSE(8.83,0,.10)+0*Vol_ileum*ka_il
vol_ij = Vol_ileum*ka_ic
Vol_jej(t) = Vol_jej(t - dt) + (voil_dj + jej_secretion - vol_ji - Adsorp_jej) * dt
INIT Vol_jej = 0

INFLOWS:
voil_dj = Vol_duod*ka_dj
jej_secretion = PULSE(2.67,0,.1)

OUTFLOWS:
vol_ji = Vol_jej*ka_ji
Adsorp_jej = PULSE(15.76,0,.1)+0*Vol_jej*ka_je
Vol_stom(t) = Vol_stom(t - dt) + (Stom_Secretion - vol_sd - Adsorp_Stom) * dt
INIT Vol_stom = PULSE(8.33,0,.1)

INFLOWS:
Stom_Secretion = PULSE(16.67,0,.1)

OUTFLOWS:
vol_sd = kf_sd*Vol_stom
Adsorp_Stom = 0*Vol_stom*ka_sd
conc_plasma = (amt_plasma/volume)*mg_to_ug
k12 = .839
k21 = .67
ka_co = 1
ka_col = 3
ka_dj = 3
ka_du = 1
ka_ic = 3
ka_il = 8.83
ka_je = 1
ka_ji = 3
ka_sd = 1
kf_sd = 2.8
k_elim = .161
mg_to_ug = 1000
perm_colon = 3.80e-6
perm_duo = 1.10e-6
perm_Il = 4.06e-6
perm_jej = 2.17e-6
perm_stom = 1.10e-6
ph_s = 1.5
ph_s_2 = 6.6
ph_s_3 = 6.6
```

```
ph_s_4 = 7.5
ph_s_5 = 6.6
SA_colon = 138
SA_duo = 125
SA_Il = 102
SA_jej = 182
SA_stom = 50
volume = 4*19200
sol_profile = GRAPH(ph_s)
(1.00, 63.0), (1.50, 25.0), (2.00, 10.0), (2.50, 5.00), (3.00, 4.00), (3.50, 3.80), (4.00, 3.65), (4.50, 3.50), (5.00, 3.65), (5.50, 3.65), (6.00, 3.65), (6.50, 3.65), (7.00, 3.65), (7.50, 3.65), (8.00, 3.65), (8.50, 4.00), (9.00, 5.00), (9.50, 12.0), (10.0, 23.5)
sol_profile_2 = GRAPH(ph_s_2)
(1.00, 63.0), (1.50, 25.0), (2.00, 10.0), (2.50, 5.00), (3.00, 4.00), (3.50, 3.80), (4.00, 3.65), (4.50, 3.50), (5.00, 3.65), (5.50, 3.65), (6.00, 3.65), (6.50, 3.65), (7.00, 3.65), (7.50, 3.65), (8.00, 3.65), (8.50, 4.00), (9.00, 5.00), (9.50, 12.0), (10.0, 23.5)
sol_profile_3 = GRAPH(ph_s_3)
(1.00, 63.0), (1.50, 25.0), (2.00, 10.0), (2.50, 5.00), (3.00, 4.00), (3.50, 3.80), (4.00, 3.65), (4.50, 3.50), (5.00, 3.65), (5.50, 3.65), (6.00, 3.65), (6.50, 3.65), (7.00, 3.65), (7.50, 3.65), (8.00, 3.65), (8.50, 4.00), (9.00, 5.00), (9.50, 12.0), (10.0, 23.5)
sol_profile_4 = GRAPH(ph_s_4)
(1.00, 63.0), (1.50, 25.0), (2.00, 10.0), (2.50, 5.00), (3.00, 4.00), (3.50, 3.80), (4.00, 3.65), (4.50, 3.50), (5.00, 3.65), (5.50, 3.65), (6.00, 3.65), (6.50, 3.65), (7.00, 3.65), (7.50, 3.65), (8.00, 3.65), (8.50, 4.00), (9.00, 5.00), (9.50, 12.0), (10.0, 23.5)
sol_profile_5 = GRAPH(ph_s_5)
(1.00, 63.0), (1.50, 25.0), (2.00, 10.0), (2.50, 5.00), (3.00, 4.00), (3.50, 3.80), (4.00, 3.65), (4.50, 3.50), (5.00, 3.65), (5.50, 3.65), (6.00, 3.65), (6.50, 3.65), (7.00, 3.65), (7.50, 3.65), (8.00, 3.65), (8.50, 4.00), (9.00, 5.00), (9.50, 12.0), (10.0, 23.5)
```

Appendix 3: Abbreviation Key For GI Model

The legend/key has been divided into sub-sections corresponding to the sub-sections of the model diagram.

Numbered suffixes (1, 2, 3, 4, 5, 6) have been assigned to distinguish between intestinal regions (stomach, duodenum, jejunum, ileum, colon, and waste, respectively).

1 – stomach
2 – duodenum
3 – jejunum
4 – ileum
5 – colon
6 – waste

For example, VOL 1 is the volume in the stomach, MASS 3 is the insoluble mass in the jejunum. In the equations, COMP 1 indicates the stomach, COMP 2 the duodenum, COMP 3, the jejunum, etc.

Ghosts are listed under the sub-section containing the original reservoir, flow regulator, or converter.

Abbreviations listed in italics are regionally dependent and set up as arrays to allow independent values for each intestinal region.

In general, ADJ as a prefix indicates a calculated parameter value (ADJ = adjusted), while ADJ as a suffix indicates an adjustment parameter (ADJ = adjustment).

Intestinal model

<u>Reservoirs/Compartments</u>

VOL ABS	Fluid volume absorbed
VOL	Fluid volume
C REL	Mass of drug contained with a formulation or controlled release device
MASS	Insoluble mass of drug (not contained within the formulation or controlled release device)
SOL	Soluble mass of drug
ABSORPTION	Mass of drug absorbed

<u>Flow regulators</u>

REABS	Rate of water absorption
VOL OUT	Fluid volume transit rate
CR OUT	Formulation or controlled release device transit rate
CR INPUT	Drug release rate from formulation or controlled release device
MASS OUT	Insoluble drug mass transit rate
DISS PRECIP	Dissolution rate
SOL OUT	Soluble drug mass transit rate
FLUX	Absorption rate

ADJ PARMS (Adjustment Parameters)

VOL ADJ	Fluid volume absorption adjustment parameter
DISS ADJ	Dissolution rate adjustment parameter
TRANSIT ADJ	Transit time adjustment parameter
SA ADJ	Surface area adjustment parameter
FLUX ADJ	Passive Absorption adjustment parameter
EFFLUX ADJ	Efflux or secretion adjustment parameter
CARRIER ADJ	Active absorption adjustment parameter

PARMS (Parameters)

VOL PARM	Fluid volume absorption rate constant
SURFACE AREA	Surface area available for absorption
DOSE	The administered dose of drug
INIT VOLUME	The administered volume of water or fluid
TIME IN HOURS	A clock
pH	The physiological pH value
PARACELLULAR	A user controlled switch used to adjust absorption based on absorption mechanism

TRANSIT TIME

TRANSFERS	GI transit rate constant
CUMU TT	Cumulative transit time
ADJ TRANSIT TIME	Adjusted GI transit time incorporating adjustment parameter and user input
USER TT INPUT	User controlled adjustments to the GI transit time

OUTPUT CALCULATIONS

ABSORBED TOTAL	Total mass of drug absorbed (sum of ABSORPTION 1...5)

FDp%	Fraction or the dose absorbed into portal vein x 100
FLUX TOTAL	Total absorption rate (sum of FLUX 1...5)
CUM DISS	Cumulative drug mass dissolved
CR Release	Cumulative drug mass released from formulation
CUM DISS RATE	Sum of DISS PRECIP 1...5
CR cumrate	Summ of CR INPUT 1...5

PERMEABILITY CALCULATION

ADJ PERM	Adjusted permeability ncorporating all transport mechanisms and relevant adjustment parameters
ACT PE	Active or carrier-mediated absorptive permeability
Km	Constant from the Michaelis-Mentin type permeability equation for active transport
REGIONAL	Passive permeability after regional correlation calculation (same as PASS PE if regional correlation is not used)
PASS PE	Passive permeability entered by user
RC	A logical function used in determining the regional correlation
RCSUM	A logical function used in determining the regional correlation

SOLUBILITY CALCULATION

USER pH	User supplied pH value for which a solubility value is available
USER SOLUB	User supplied solubility value corresponding to the USER pH value
ADJ SOLUB	Solubility calculated (if necessary) at the appropriate pH value using the entered USER pH and USER SOLUB values

CONTROLLED RELEASE CALCULATION

CR RATE	The instantaneous release rate from the formulation
CR DOSE	The total dose contained with the formulation
CR AT TIME	The cumulative drug mass release profile
CR AT LAST	The cumulative drug mass release profile

Note: CR AT TIME holds the value at the current time value (t), CR AT LAST holds the value at the immediately preceeding time value (t-1)

CONC CALCULATION

CONCENTRATIONS	The dissolved drug concentration

DISSOLUTION CALCULATION

PRECIP	Precipitation rate constant
DISSOL	Dissolution rate constant
ADJ DISS PRECIP	Adjusted rate constant incorporating PRECIP, DISSOL and calculated concentration

Appendix 4: Equations, Parameters and Values For GI Model

ADJ PARMS

- CARRIER_ADJ[COMPS] = 0
- DISS_ADJ[COMP_1] = 1
- DISS_ADJ[COMP_2] = 1
- DISS_ADJ[COMP_3] = 1
- DISS_ADJ[COMP_4] = 1
- DISS_ADJ[COMP_5] = 1
- EFFLUX_ADJ[COMPS] = 1
- FLUX_ADJ[COMP_1] = 1
- FLUX_ADJ[COMP_2] = 10
- FLUX_ADJ[COMP_3] = 8
- FLUX_ADJ[COMP_4] = 2
- FLUX_ADJ[COMP_5] = 1
- SA_ADJ[COMP_1] = 1
- SA_ADJ[COMP_2] = 1
- SA_ADJ[COMP_3] = 1
- SA_ADJ[COMP_4] = 1
- SA_ADJ[COMP_5] = 1
- TRANSIT_ADJ[COMP_1] = 1
- TRANSIT_ADJ[COMP_2] = 1
- TRANSIT_ADJ[COMP_3] = 1
- TRANSIT_ADJ[COMP_4] = 1
- TRANSIT_ADJ[COMP_5] = 1
- VOL_ADJ[COMP_1] = 1
- VOL_ADJ[COMP_2] = 1
- VOL_ADJ[COMP_3] = 1
- VOL_ADJ[COMP_4] = 1
- VOL_ADJ[COMP_5] = 1

CONC CALCULATION

- CONCENTRATIONS[COMP_1] = if VOL_1=0.0 then 0 else if ADJ_SOLUB[COMP_1]<SOL_1/VOL_1 then ADJ_SOLUB[COMP_1] else SOL_1/VOL_1 + 0*(SOL_2+SOL_5+SOL_3+SOL_4+VOL_3+VOL_2+VOL_4+VOL_5)
- CONCENTRATIONS[COMP_2] = if VOL_2 = 0.0 then 0 else if (VOL_2<1e-6 AND SOL_2<1e-7) then 0 else if ADJ_SOLUB[COMP_2]<SOL_2/VOL_2 then ADJ_SOLUB[COMP_2] else SOL_2/VOL_2 +0*(SOL_1+SOL_5+SOL_3+SOL_4+VOL_3+VOL_1+VOL_5+VOL_4)
- CONCENTRATIONS[COMP_3] = if VOL_3 = 0.0 then 0 else if (VOL_3<1e-6 AND SOL_3<1e-7) then 0 else if ADJ_SOLUB[COMP_3]<SOL_3/VOL_3 then ADJ_SOLUB[COMP_3] else SOL_3/VOL_3 +0*(SOL_1+SOL_2+SOL_4+SOL_5+VOL_5+VOL_4+VOL_1+VOL_2)
- CONCENTRATIONS[COMP_4] = if VOL_4 = 0.0 then 0 else if (VOL_4<1e-6 AND SOL_4<1e-7) then 0 else if ADJ_SOLUB[COMP_4]<SOL_4/VOL_4 then ADJ_SOLUB[COMP_4] else SOL_4/VOL_4 +0*(SOL_1+SOL_2+SOL_3+SOL_5+VOL_1+VOL_2+VOL_3+VOL_5)

CONCENTRATIONS[COMP_5] = if VOL_5 = 0.0 then 0 else if (VOL_5<1e-6 AND SOL_5<1e-7) then 0 else if ADJ_SOLUB[COMP_5]<SOL_5/VOL_5 then ADJ_SOLUB[COMP_5] else SOL_5/VOL_5
+0*(SOL_1+SOL_4+SOL_3+SOL_2+VOL_3+VOL_1+VOL_2+VOL_4)

CONTROL RELEASE CALCULATION

CR_DOSE = 0

CR_RATE = (CR_AT_TIME-CR_AT_LAST)*20*(CR_DOSE/100)

CR_AT_LAST = GRAPH(TIME-DT)
(0.00, 0.00), (0.25, 17.7), (0.5, 31.5), (0.75, 42.2), (1.00, 50.6), (1.25, 57.1), (1.50, 62.1), (1.75, 66.1), (2.00, 69.2), (2.25, 71.6), (2.50, 73.4), (2.75, 74.9), (3.00, 76.0), (3.25, 76.9), (3.50, 77.6), (3.75, 78.1), (4.00, 78.5), (4.25, 78.9), (4.50, 79.1), (4.75, 79.3), (5.00, 79.5), (5.25, 79.6), (5.50, 79.7), (5.75, 79.7), (6.00, 79.8), (6.25, 79.8), (6.50, 79.9), (6.75, 79.9), (7.00, 79.9), (7.25, 79.9), (7.50, 80.0), (7.75, 80.0), (8.00, 80.0), (8.25, 80.0), (8.50, 80.0), (8.75, 80.0), (9.00, 80.0), (9.25, 80.0), (9.50, 80.0), (9.75, 80.0), (10.0, 80.0), (10.3, 80.0), (10.5, 80.0), (10.8, 80.0), (11.0, 80.0), (11.3, 80.0), (11.5, 80.0), (11.8, 80.0), (12.0, 80.0), (12.3, 80.0), (12.5, 80.0), (12.8, 80.0), (13.0, 80.0)...

CR_AT_TIME = GRAPH(TIME)
(0.00, 0.00), (0.25, 17.7), (0.5, 31.5), (0.75, 42.2), (1.00, 50.6), (1.25, 57.1), (1.50, 62.1), (1.75, 66.1), (2.00, 69.2), (2.25, 71.6), (2.50, 73.4), (2.75, 74.9), (3.00, 76.0), (3.25, 76.9), (3.50, 77.6), (3.75, 78.1), (4.00, 78.5), (4.25, 78.9), (4.50, 79.1), (4.75, 79.3), (5.00, 79.5), (5.25, 79.6), (5.50, 79.7), (5.75, 79.7), (6.00, 79.8), (6.25, 79.8), (6.50, 79.9), (6.75, 79.9), (7.00, 79.9), (7.25, 79.9), (7.50, 80.0), (7.75, 80.0), (8.00, 80.0), (8.25, 80.0), (8.50, 80.0), (8.75, 80.0), (9.00, 80.0), (9.25, 80.0), (9.50, 80.0), (9.75, 80.0), (10.0, 80.0), (10.3, 80.0), (10.5, 80.0), (10.8, 80.0), (11.0, 80.0), (11.3, 80.0), (11.5, 80.0), (11.8, 80.0), (12.0, 80.0), (12.3, 80.0), (12.5, 80.0), (12.8, 80.0), (13.0, 80.0)...

DISSOLUTION CALCULATION

ADJ_DISS_PRECIP[COMP_1] = if VOL_1=0 then 0 else if (SOL_1/VOL_1<ADJ_SOLUB[COMP_1]) then (DISSOL[COMP_1]*DISS_ADJ[COMP_1]*MASS_1*(ADJ_SOLUB[COMP_1]-SOL_1/VOL_1)) else ((SOL_1/VOL_1)-ADJ_SOLUB[COMP_1])*PRECIP[COMP_1]+
0*(MASS_1+MASS_2+MASS_3+MASS_4+MASS_5+SOL_1+SOL_2+SOL_3+SOL_4+SOL_5+VOL_1+VOL_2+VOL_3+VOL_4+VOL_5)

ADJ_DISS_PRECIP[COMP_2] = if VOL_2=0 then 0 else if (SOL_2/VOL_2<ADJ_SOLUB[COMP_2]) then (DISSOL[COMP_2]*DISS_ADJ[COMP_2]*MASS_2*(ADJ_SOLUB[COMP_2]-SOL_2/VOL_2)) else ((SOL_2/VOL_2)-ADJ_SOLUB[COMP_2])*PRECIP[COMP_2]
+0*(MASS_1+MASS_2+MASS_3+MASS_4+MASS_5+SOL_1+SOL_2+SOL_3+SOL_4+SOL_5+VOL_1+VOL_2+VOL_3+VOL_4+VOL_5)

ADJ_DISS_PRECIP[COMP_3] = if VOL_3=0 then 0 else if (SOL_3/VOL_3<ADJ_SOLUB[COMP_3]) then (DISSOL[COMP_3]*DISS_ADJ[COMP_3]*MASS_3*(ADJ_SOLUB[COMP_3]-SOL_3/VOL_3)) else ((SOL_3/VOL_3)-ADJ_SOLUB[COMP_3])*PRECIP[COMP_3]
+0*(MASS_1+MASS_2+MASS_3+MASS_4+MASS_5+SOL_1+SOL_2+SOL_3+SOL_4+SOL_5+VOL_1+VOL_2+VOL_3+VOL_4+VOL_5)

```
ADJ_DISS_PRECIP[COMP_4] = if VOL_4=0 then 0 else if
(SOL_4/VOL_4<ADJ_SOLUB[COMP_4]) then
(DISSOL[COMP_4]*DISS_ADJ[COMP_4]*MASS_4*(ADJ_SOLUB[COMP_4]-SOL_4/VOL_4)) else
((SOL_4/VOL_4)-ADJ_SOLUB[COMP_4])*PRECIP[COMP_4]
+0*(MASS_1+MASS_2+MASS_3+MASS_4+MASS_5+SOL_1+SOL_2+SOL_3+SOL_4+SOL_5+V
OL_1+VOL_2+VOL_3+VOL_4+VOL_5)
ADJ_DISS_PRECIP[COMP_5] = if VOL_5=0 then 0 else if
(SOL_5/VOL_5<ADJ_SOLUB[COMP_5]) then
(DISSOL[COMP_5]*DISS_ADJ[COMP_5]*MASS_5*(ADJ_SOLUB[COMP_5]-SOL_5/VOL_5)) else
((SOL_5/VOL_5)-ADJ_SOLUB[COMP_5])*PRECIP[COMP_5]
+0*(MASS_1+MASS_2+MASS_3+MASS_4+MASS_5+SOL_1+SOL_2+SOL_3+SOL_4+SOL_5+V
OL_1+VOL_2+VOL_3+VOL_4+VOL_5)
DISSOL[COMP_1] = 1
DISSOL[COMP_2] = 1
DISSOL[COMP_3] = 1
DISSOL[COMP_4] = 1
DISSOL[COMP_5] = 1
PRECIP[COMP_1] = 10
PRECIP[COMP_2] = 10
PRECIP[COMP_3] = 10
PRECIP[COMP_4] = 10
PRECIP[COMP_5] = 10
INPUTS
INTESTINAL MODEL
ABSORPTION_1(t) = ABSORPTION_1(t - dt) + (FLUX_1) * dt
INIT ABSORPTION_1 = 0
 INFLOWS:
   FLUX_1 =
   CONCENTRATIONS[COMP_1]*ADJ_PERM[COMP_1]*SURFACE_AREA[COMP_1]

ABSORPTION_2(t) = ABSORPTION_2(t - dt) + (FLUX_2) * dt
INIT ABSORPTION_2 = 0
 INFLOWS:
   FLUX_2 =
   CONCENTRATIONS[COMP_2]*ADJ_PERM[COMP_2]*SURFACE_AREA[COMP_2]

ABSORPTION_3(t) = ABSORPTION_3(t - dt) + (FLUX_3) * dt
INIT ABSORPTION_3 = 0
 INFLOWS:
   FLUX_3 =
   CONCENTRATIONS[COMP_3]*ADJ_PERM[COMP_3]*SURFACE_AREA[COMP_3]

ABSORPTION_4(t) = ABSORPTION_4(t - dt) + (FLUX_4) * dt
INIT ABSORPTION_4 = 0
 INFLOWS:
```

```
FLUX_4 =
CONCENTRATIONS[COMP_4]*ADJ_PERM[COMP_4]*SURFACE_AREA[COMP_4]

ABSORPTION_5(t) = ABSORPTION_5(t - dt) + (FLUX_5) * dt
INIT ABSORPTION_5 = 0
 INFLOWS:
   FLUX_5 = if time<32 then
   CONCENTRATIONS[COMP_5]*ADJ_PERM[COMP_5]*SURFACE_AREA[COMP_5]*(32-ti
   me)/48*(VOL_5/17.2) else 0
C_REL_1(t) = C_REL_1(t - dt) + (- CR_OUT_1 - CR_INPUT_1) * dt
INIT C_REL_1 = CR_DOSE
 OUTFLOWS:
   CR_OUT_1 = IF TIME >= CUMU_TT[COMP_1] THEN C_REL_1*10000 ELSE 0
   CR_INPUT_1 = if TIME>CUMU_TT[COMP_1] then 0 else CR_RATE
C_REL_2(t) = C_REL_2(t - dt) + (CR_OUT_1 - CR_OUT_2 - CR_INPUT_2) * dt
INIT C_REL_2 = 0
 INFLOWS:
   CR_OUT_1 = IF TIME >= CUMU_TT[COMP_1] THEN C_REL_1*10000 ELSE 0
 OUTFLOWS:
   CR_OUT_2 = IF TIME >= CUMU_TT[COMP_2] THEN C_REL_2*10000 ELSE 0
   CR_INPUT_2 = if TIME>CUMU_TT[COMP_2] then 0 else CR_RATE
C_REL_3(t) = C_REL_3(t - dt) + (CR_OUT_2 - CR_OUT_3 - CR_INPUT_3) * dt
INIT C_REL_3 = 0
 INFLOWS:
   CR_OUT_2 = IF TIME >= CUMU_TT[COMP_2] THEN C_REL_2*10000 ELSE 0
 OUTFLOWS:
   CR_OUT_3 = IF TIME >= CUMU_TT[COMP_3] THEN C_REL_3*10000 ELSE 0
   CR_INPUT_3 = if TIME > CUMU_TT[COMP_3] then 0 else CR_RATE
C_REL_4(t) = C_REL_4(t - dt) + (CR_OUT_3 - CR_OUT_4 - CR_INPUT_4) * dt
INIT C_REL_4 = 0
 INFLOWS:
   CR_OUT_3 = IF TIME >= CUMU_TT[COMP_3] THEN C_REL_3*10000 ELSE 0
 OUTFLOWS:
   CR_OUT_4 = IF TIME >= CUMU_TT[COMP_4] THEN C_REL_4*10000 ELSE 0
   CR_INPUT_4 = if TIME>CUMU_TT[COMP_4] then 0 else CR_RATE
C_REL_5(t) = C_REL_5(t - dt) + (CR_OUT_4 - CR_OUT_5 - CR_INPUT_5) * dt
INIT C_REL_5 = 0
 INFLOWS:
   CR_OUT_4 = IF TIME >= CUMU_TT[COMP_4] THEN C_REL_4*10000 ELSE 0
 OUTFLOWS:
   CR_OUT_5 = IF TIME >= CUMU_TT[COMP_5] THEN C_REL_5*10000 ELSE 0
   CR_INPUT_5 = if TIME>CUMU_TT[COMP_5] then 0 else CR_RATE
C_REL_6(t) = C_REL_6(t - dt) + (CR_OUT_5) * dt
INIT C_REL_6 = 0
 INFLOWS:
   CR_OUT_5 = IF TIME >= CUMU_TT[COMP_5] THEN C_REL_5*10000 ELSE 0
```

MASS_1(t) = MASS_1(t - dt) + (CR_INPUT_1 - MASS_OUT_1 - DISS_PRECIP_1) * dt
INIT MASS_1 = DOSE
INFLOWS:
CR_INPUT_1 = if TIME>CUMU_TT[COMP_1] then 0 else CR_RATE
OUTFLOWS:
MASS_OUT_1 = MASS_1*TRANSFERS[COMP_1]
DISS_PRECIP_1 = ADJ_DISS_PRECIP[COMP_1]

MASS_2(t) = MASS_2(t - dt) + (MASS_OUT_1 + CR_INPUT_2 - MASS_OUT_2 - DISS_PRECIP_2) * dt
INIT MASS_2 = 0
INFLOWS:
MASS_OUT_1 = MASS_1*TRANSFERS[COMP_1]
CR_INPUT_2 = if TIME>CUMU_TT[COMP_2] then 0 else CR_RATE
OUTFLOWS:
MASS_OUT_2 = MASS_2*TRANSFERS[COMP_2]
DISS_PRECIP_2 = ADJ_DISS_PRECIP[COMP_2]

MASS_3(t) = MASS_3(t - dt) + (CR_INPUT_3 + MASS_OUT_2 - MASS_OUT_3 - DISS_PRECIP_3) * dt
INIT MASS_3 = 0
INFLOWS:
CR_INPUT_3 = if TIME > CUMU_TT[COMP_3] then 0 else CR_RATE
MASS_OUT_2 = MASS_2*TRANSFERS[COMP_2]
OUTFLOWS:
MASS_OUT_3 = MASS_3*TRANSFERS[COMP_3]
DISS_PRECIP_3 = ADJ_DISS_PRECIP[COMP_3]

MASS_4(t) = MASS_4(t - dt) + (CR_INPUT_4 + MASS_OUT_3 - MASS_OUT_4 - DISS_PRECIP_4) * dt
INIT MASS_4 = 0
INFLOWS:
CR_INPUT_4 = if TIME>CUMU_TT[COMP_4] then 0 else CR_RATE
MASS_OUT_3 = MASS_3*TRANSFERS[COMP_3]
OUTFLOWS:
MASS_OUT_4 = MASS_4*TRANSFERS[COMP_4]
DISS_PRECIP_4 = ADJ_DISS_PRECIP[COMP_4]

MASS_5(t) = MASS_5(t - dt) + (CR_INPUT_5 + MASS_OUT_4 - MASS_OUT_5 - DISS_PRECIP_5) * dt
INIT MASS_5 = 0
INFLOWS:
CR_INPUT_5 = if TIME>CUMU_TT[COMP_5] then 0 else CR_RATE
MASS_OUT_4 = MASS_4*TRANSFERS[COMP_4]
OUTFLOWS:
MASS_OUT_5 = if time>4 then MASS_5*TRANSFERS[COMP_5] else 0
DISS_PRECIP_5 = ADJ_DISS_PRECIP[COMP_5]

MASS_6(t) = MASS_6(t - dt) + (MASS_OUT_5) * dt
INIT MASS_6 = 0
INFLOWS:

```
MASS_OUT_5 = if time>4 then MASS_5*TRANSFERS[COMP_5] else 0
SOL_1(t) = SOL_1(t - dt) + (DISS_PRECIP_1 - SOL_OUT_1 - FLUX_1) * dt
INIT SOL_1 = 0
 INFLOWS:
   DISS_PRECIP_1 = ADJ_DISS_PRECIP[COMP_1]
 OUTFLOWS:
   SOL_OUT_1 = SOL_1*TRANSFERS[COMP_1]
   FLUX_1 =
   CONCENTRATIONS[COMP_1]*ADJ_PERM[COMP_1]*SURFACE_AREA[COMP_1]
SOL_2(t) = SOL_2(t - dt) + (SOL_OUT_1 + DISS_PRECIP_2 - SOL_OUT_2 - FLUX_2) * dt
INIT SOL_2 = 0

 INFLOWS:
   SOL_OUT_1 = SOL_1*TRANSFERS[COMP_1]
   DISS_PRECIP_2 = ADJ_DISS_PRECIP[COMP_2]
 OUTFLOWS:
   SOL_OUT_2 = SOL_2*TRANSFERS[COMP_2]
   FLUX_2 =
   CONCENTRATIONS[COMP_2]*ADJ_PERM[COMP_2]*SURFACE_AREA[COMP_2]
SOL_3(t) = SOL_3(t - dt) + (DISS_PRECIP_3 + SOL_OUT_2 - SOL_OUT_3 - FLUX_3) * dt
INIT SOL_3 = 0

 INFLOWS:
   DISS_PRECIP_3 = ADJ_DISS_PRECIP[COMP_3]
   SOL_OUT_2 = SOL_2*TRANSFERS[COMP_2]
 OUTFLOWS:
   SOL_OUT_3 = SOL_3*TRANSFERS[COMP_3]
   FLUX_3 =
   CONCENTRATIONS[COMP_3]*ADJ_PERM[COMP_3]*SURFACE_AREA[COMP_3]
SOL_4(t) = SOL_4(t - dt) + (DISS_PRECIP_4 + SOL_OUT_3 - SOL_OUT_4 - FLUX_4) * dt
INIT SOL_4 = 0

 INFLOWS:
   DISS_PRECIP_4 = ADJ_DISS_PRECIP[COMP_4]
   SOL_OUT_3 = SOL_3*TRANSFERS[COMP_3]
 OUTFLOWS:
   SOL_OUT_4 = SOL_4*TRANSFERS[COMP_4]
   FLUX_4 =
   CONCENTRATIONS[COMP_4]*ADJ_PERM[COMP_4]*SURFACE_AREA[COMP_4]
SOL_5(t) = SOL_5(t - dt) + (DISS_PRECIP_5 + SOL_OUT_4 - SOL_OUT_5 - FLUX_5) * dt
INIT SOL_5 = 0

 INFLOWS:
```

```
    DISS_PRECIP_5 = ADJ_DISS_PRECIP[COMP_5]
    SOL_OUT_4 = SOL_4*TRANSFERS[COMP_4]
  OUTFLOWS:
    SOL_OUT_5 = if time>4 then SOL_5*TRANSFERS[COMP_5] else 0
    FLUX_5 = if time<32 then
    CONCENTRATIONS[COMP_5]*ADJ_PERM[COMP_5]*SURFACE_AREA[COMP_5]*(32-ti
    me)/48*(VOL_5/17.2) else 0
SOL_6(t) = SOL_6(t - dt) + (SOL_OUT_5) * dt
INIT SOL_6 = 0
  INFLOWS:
    SOL_OUT_5 = if time>4 then SOL_5*TRANSFERS[COMP_5] else 0
VOL_1(t) = VOL_1(t - dt) + (- REABS_1 - VOL_OUT_1) * dt
INIT VOL_1 = INIT_VOLUME
  OUTFLOWS:
    REABS_1 = VOL_1*VOL_PARM[COMP_1]
    VOL_OUT_1 = VOL_1*TRANSFERS[COMP_1]
VOL_2(t) = VOL_2(t - dt) + (VOL_OUT_1 - VOL_OUT_2 - REABS_2) * dt
INIT VOL_2 = 0
  INFLOWS:
    VOL_OUT_1 = VOL_1*TRANSFERS[COMP_1]
  OUTFLOWS:
    VOL_OUT_2 = VOL_2*TRANSFERS[COMP_2]
    REABS_2 = VOL_2*VOL_PARM[COMP_2]
VOL_3(t) = VOL_3(t - dt) + (VOL_OUT_2 - VOL_OUT_3 - REABS_3) * dt
INIT VOL_3 = 0
  INFLOWS:
    VOL_OUT_2 = VOL_2*TRANSFERS[COMP_2]
  OUTFLOWS:
    VOL_OUT_3 = VOL_3*TRANSFERS[COMP_3]
    REABS_3 = VOL_3*VOL_PARM[COMP_3]
VOL_4(t) = VOL_4(t - dt) + (VOL_OUT_3 - VOL_OUT_4 - REABS_4) * dt
INIT VOL_4 = 0
  INFLOWS:
    VOL_OUT_3 = VOL_3*TRANSFERS[COMP_3]
  OUTFLOWS:
    VOL_OUT_4 = VOL_4*TRANSFERS[COMP_4]
    REABS_4 = VOL_4*VOL_PARM[COMP_4]
VOL_5(t) = VOL_5(t - dt) + (VOL_OUT_4 - VOL_OUT_5 - REABS_5) * dt
INIT VOL_5 = 0
  INFLOWS:
    VOL_OUT_4 = VOL_4*TRANSFERS[COMP_4]
  OUTFLOWS:
    VOL_OUT_5 = VOL_5*TRANSFERS[COMP_5]
    REABS_5 = VOL_5*VOL_PARM[COMP_5]
VOL_6(t) = VOL_6(t - dt) + (VOL_OUT_5) * dt
INIT VOL_6 = 0
```

```
INFLOWS:
  VOL_OUT_5 = VOL_5*TRANSFERS[COMP_5]
VOL_ABS_1(t) = VOL_ABS_1(t - dt) + (REABS_1) * dt
INIT VOL_ABS_1 = 0
 INFLOWS:
  REABS_1 = VOL_1*VOL_PARM[COMP_1]
VOL_ABS_2(t) = VOL_ABS_2(t - dt) + (REABS_2) * dt
INIT VOL_ABS_2 = 0
 INFLOWS:
  REABS_2 = VOL_2*VOL_PARM[COMP_2]
VOL_ABS_3(t) = VOL_ABS_3(t - dt) + (REABS_3) * dt
INIT VOL_ABS_3 = 0
 INFLOWS:
  REABS_3 = VOL_3*VOL_PARM[COMP_3]
VOL_ABS_4(t) = VOL_ABS_4(t - dt) + (REABS_4) * dt
INIT VOL_ABS_4 = 0
 INFLOWS:
  REABS_4 = VOL_4*VOL_PARM[COMP_4]
VOL_ABS_5(t) = VOL_ABS_5(t - dt) + (REABS_5) * dt
INIT VOL_ABS_5 = 0
 INFLOWS:
  REABS_5 = VOL_5*VOL_PARM[COMP_5]
MULTI DOSE CALCULATION
OUTPUT CALCULATIONS
CR_Release(t) = CR_Release(t - dt) + (CR_cumrate) * dt
INIT CR_Release = 0
 INFLOWS:
  CR_cumrate = CR_INPUT_1+CR_INPUT_2+CR_INPUT_3+CR_INPUT_4+CR_INPUT_5
CUM_DISS(t) = CUM_DISS(t - dt) + (CUMM_DISS_RATE) * dt
INIT CUM_DISS = 0
 INFLOWS:
  CUMM_DISS_RATE =
  DISS_PRECIP_1+DISS_PRECIP_2+DISS_PRECIP_3+DISS_PRECIP_4+DISS_PRECIP
  5
ABSORBED_TOTAL = ABSORPTION_2+ABSORPTION_3+ABSORPTION_4+ABSORPTION_5
FDp% = ABSORBED_TOTAL/DOSE*100
FLUX_TOTAL = FLUX_2+FLUX_3+FLUX_4+FLUX_5
PARMS
DOSE = 1000
INIT_VOLUME = 100
PARACELLULAR = 1
pH[COMP_1] = 1.5
pH[COMP_2] = 5
pH[COMP_3] = 6.5
```

pH[COMP_4] = 7
pH[COMP_5] = 6.5
SURFACE_AREA[COMP_1] = if PARACELLULAR =0 then 50*SA_ADJ[COMP_1] else 50*SA_ADJ[COMP_1]
SURFACE_AREA[COMP_2] = if PARACELLULAR=0 then 150*SA_ADJ[COMP_2] else 7.5*SA_ADJ[COMP_2]
SURFACE_AREA[COMP_3] = if PARACELLULAR=0 then 1000*SA_ADJ[COMP_3] else 50*SA_ADJ[COMP_3]
SURFACE_AREA[COMP_4] = if PARACELLULAR=0 then 1000*SA_ADJ[COMP_4] else 50*SA_ADJ[COMP_4]
SURFACE_AREA[COMP_5] = if PARACELLULAR=0 then 850*SA_ADJ[COMP_5] else 42.5*SA_ADJ[COMP_5]
TIME_IN_HOURS = TIME
VOL_PARM[COMP_1] = 0*VOL_ADJ[COMP_1]
VOL_PARM[COMP_2] = 0*VOL_ADJ[COMP_2]
VOL_PARM[COMP_3] = 1.75*VOL_ADJ[COMP_3]
VOL_PARM[COMP_4] = 1.75*VOL_ADJ[COMP_4]
VOL_PARM[COMP_5] = 0.10*VOL_ADJ[COMP_5]

PERMEABILITY CALCULATION

ACT_PE[COMPS] = [0 ,
0 ,
0 ,
0 ,
0]

ADJ_PERM[COMP_1] = (2/(1+EFFLUX_ADJ[COMP_1]))*REGIONAL[COMP_1]*FLUX_ADJ[COMP_1]*3600+(CARRIER_DJ[COMP_1]*ACT_PE[COMP_1]*3600/(1+(CONCENTRATIONS[COMP_1]/(Km[COMP_1]))))*0

ADJ_PERM[COMP_2] = (2/(1+EFFLUX_ADJ[COMP_2]))*REGIONAL[COMP_2]*FLUX_ADJ[COMP_2]*3600+(CARRIER_DJ[COMP_2]*ACT_PE[COMP_2]*3600/(1+(CONCENTRATIONS[COMP_2]/(Km[COMP_2]))))

ADJ_PERM[COMP_3] = (2/(1+EFFLUX_ADJ[COMP_3]))*REGIONAL[COMP_3]*FLUX_ADJ[COMP_3]*3600+(CARRIER_DJ[COMP_3]*ACT_PE[COMP_3]*3600/(1+(CONCENTRATIONS[COMP_3]/(Km[COMP_3]))))

ADJ_PERM[COMP_4] = (2/(1+EFFLUX_ADJ[COMP_4]))*REGIONAL[COMP_4]*FLUX_ADJ[COMP_4]*3600+(CARRIER_DJ[COMP_4]*ACT_PE[COMP_4]*3600/(1+(CONCENTRATIONS[COMP_4]/(Km[COMP_4]))))

ADJ_PERM[COMP_5] = (2/(1+EFFLUX_ADJ[COMP_5]))*REGIONAL[COMP_5]*FLUX_ADJ[COMP_5]*3600+(CARRIER_DJ[COMP_5]*ACT_PE[COMP_5]*3600/(1+(CONCENTRATIONS[COMP_5]/(Km[COMP_5]))))

```
Km[COMPS] = [1 ,
1 ,
1 ,
1 ,
1 ]
PASS_PE[COMPS] = [0 ,
1.10E-06 ,
2.17E-06 ,
4.06E-06 ,
3.80E-06 ]
RC[COMP_1] = PASS_PE[COMP_1]*0
RC[COMP_2] = IF PASS_PE[COMP_2]>0 THEN 1 ELSE 0
RC[COMP_3] = IF PASS_PE[COMP_3]>0 THEN 2 ELSE 0
RC[COMP_4] = IF PASS_PE[COMP_4]>0 THEN 4 ELSE 0
RC[COMP_5] = PASS_PE[COMP_5]*0
RCSUM = RC[COMP_2]+RC[COMP_3]+RC[COMP_4]
REGIONAL[COMP_1] = PASS_PE[COMP_1]+RCSUM*0
REGIONAL[COMP_2] = if RCSUM=2 then
(EXP( -9.011926 + 2.594378 *LOGN(1/PASS_PE[COMP_2]) -0.065515
*LOGN(1/PASS_PE[COMP_2])^2))^(-1) else
if RCSUM=4 then

(EXP(-0.369414*LOGN(1/PASS_PE[COMP_4])+0.23756*LOGN(1/PASS_PE[COMP_4])^2-0.009
9719*LOGN(1/PASS_PE[COMP_4])^3))^(-1) else
if RCSUM=6 then
0.5*(EXP( -9.011926 + 2.594378 *LOGN(1/PASS_PE[COMP_3]) -0.065515
*LOGN(1/PASS_PE[COMP_3])^2))^(-1)
+0.5*(EXP( -21.009845 + 4.544238 *LOGN(1/PASS_PE[COMP_4]) -0.140815
*LOGN(1/PASS_PE[COMP_4])^2))^(-1) else
PASS_PE[COMP_2]
REGIONAL[COMP_3] = if RCSUM=1 then
(EXP( -3.238469 + 1.509131 *LOGN(1/PASS_PE[COMP_2]) -0.022109
*LOGN(1/PASS_PE[COMP_2])^2))^(-1) else
if RCSUM=4 then

(EXP(-0.093739*LOGN(1/PASS_PE[COMP_4])+0.182344*LOGN(1/PASS_PE[COMP_4])^2-0.00
23631*LOGN(1/PASS_PE[COMP_4])^3))^(-1) else
if RCSUM=5 then
0.5*(EXP( -3.238469 + 1.509131 *LOGN(1/PASS_PE[COMP_2]) -0.022109
*LOGN(1/PASS_PE[COMP_2])^2))^(-1)
+0.5*(EXP( -15.415683 + 3.543563 *LOGN(1/PASS_PE[COMP_4]) -0.100318
*LOGN(1/PASS_PE[COMP_4])^2))^(-1) else
PASS_PE[COMP_3]
```

```
REGIONAL[COMP_4] = if RCSUM=1 then
(EXP( 14.455255 -1.264630 *LOGN(1/PASS_PE[COMP_2]) + 0.082015
*LOGN(1/PASS_PE[COMP_2])^2))^(-1) else
if RCSUM=2 then
(EXP( 11.480333 -0.791109 *LOGN(1/PASS_PE[COMP_3]) + 0.066063
*LOGN(1/PASS_PE[COMP_3])^2))^(-1) else
if RCSUM=3 then
0.5*(EXP( 14.455255 -1.264630 *LOGN(1/PASS_PE[COMP_2]) + 0.082015
*LOGN(1/PASS_PE[COMP_2])^2))^(-1)
+0.5*(EXP( 11.480333 -0.791109 *LOGN(1/PASS_PE[COMP_3]) + 0.066063
*LOGN(1/PASS_PE[COMP_3])^2))^(-1) else
PASS_PE[COMP_4]
REGIONAL[COMP_5] = PASS_PE[COMP_5] +RCSUM*0
SOLUBILIY CALCULATION
ADJ_SOLUB[COMP_1] = if USER_pH[COMP_1]>=pH[COMP_1] then USER_SOLUB[COMP_1]
else
((USER_SOLUB[COMP_2]-USER_SOLUB[COMP_1])/(USER_pH[COMP_2]-USER_pH[COMP_1]
))*(pH[COMP_1]-USER_pH[COMP_1])+USER_SOLUB[COMP_1]
ADJ_SOLUB[COMP_2] = if USER_pH[COMP_2]=pH[COMP_2] then USER_SOLUB[COMP_2]
else if USER_pH[COMP_2]<pH[COMP_2] then
((USER_SOLUB[COMP_3]-USER_SOLUB[COMP_2])/(USER_pH[COMP_3]-USER_pH[COMP_2]
))*(pH[COMP_2]-USER_pH[COMP_2])+USER_SOLUB[COMP_2] else
((USER_SOLUB[COMP_2]-USER_SOLUB[COMP_1])/(USER_pH[COMP_2]-USER_pH[COMP_1]
))*(pH[COMP_2]-USER_pH[COMP_1])+USER_SOLUB[COMP_1]
ADJ_SOLUB[COMP_3] = if USER_pH[COMP_3]=pH[COMP_3] then USER_SOLUB[COMP_3]
else if USER_pH[COMP_3]<pH[COMP_3] then
((USER_SOLUB[COMP_4]-USER_SOLUB[COMP_3])/(USER_pH[COMP_4]-USER_pH[COMP_3]
))*(pH[COMP_3]-USER_pH[COMP_3])+USER_SOLUB[COMP_3] else
((USER_SOLUB[COMP_3]-USER_SOLUB[COMP_2])/(USER_pH[COMP_3]-USER_pH[COMP_2]
))*(pH[COMP_3]-USER_pH[COMP_2])+USER_SOLUB[COMP_2]
ADJ_SOLUB[COMP_4] = if USER_pH[COMP_4]=pH[COMP_4] then USER_SOLUB[COMP_4]
else if USER_pH[COMP_4]<pH[COMP_4] then
((USER_SOLUB[COMP_5]-USER_SOLUB[COMP_4])/(USER_pH[COMP_5]-USER_pH[COMP_4]
))*(pH[COMP_4]-USER_pH[COMP_4])+USER_SOLUB[COMP_4] else
((USER_SOLUB[COMP_4]-USER_SOLUB[COMP_3])/(USER_pH[COMP_4]-USER_pH[COMP_3]
))*(pH[COMP_4]-USER_pH[COMP_3])+USER_SOLUB[COMP_3]
ADJ_SOLUB[COMP_5] = if USER_pH[COMP_3]=pH[COMP_3] then USER_SOLUB[COMP_3]
else if USER_pH[COMP_3]<pH[COMP_3] then
((USER_SOLUB[COMP_4]-USER_SOLUB[COMP_3])/(USER_pH[COMP_4]-USER_pH[COMP_3]
))*(pH[COMP_3]-USER_pH[COMP_3])+USER_SOLUB[COMP_3] else
((USER_SOLUB[COMP_3]-USER_SOLUB[COMP_2])/(USER_pH[COMP_3]-USER_pH[COMP_2]
))*(pH[COMP_3]-USER_pH[COMP_2])+USER_SOLUB[COMP_2]
USER_pH[COMPS] = [1.5 ,
5 ,
6.5 ,
7 ,
7.5 ]
```

```
USER_SOLUB[COMPS] = [31 ,
3.65 ,
3.65 ,
3.65 ,
3.65 ]
TRANSIT TIME
ADJ_TRANSIT_TIME[COMP_1] = .5*TRANSIT_ADJ[COMP_1]*USER_TT_INPUT
ADJ_TRANSIT_TIME[COMP_2] = .25*TRANSIT_ADJ[COMP_2]*USER_TT_INPUT
ADJ_TRANSIT_TIME[COMP_3] = 1.5*TRANSIT_ADJ[COMP_3]*USER_TT_INPUT
ADJ_TRANSIT_TIME[COMP_4] = 1.5*TRANSIT_ADJ[COMP_4]*USER_TT_INPUT
ADJ_TRANSIT_TIME[COMP_5] = 24*TRANSIT_ADJ[COMP_5]*USER_TT_INPUT
CUMU_TT[COMP_1] = ADJ_TRANSIT_TIME[COMP_1]
CUMU_TT[COMP_2] = ADJ_TRANSIT_TIME[COMP_1]+ADJ_TRANSIT_TIME[COMP_2]
CUMU_TT[COMP_3] =
ADJ_TRANSIT_TIME[COMP_1]+ADJ_TRANSIT_TIME[COMP_2]+ADJ_TRANSIT_TIME[COMP_
3]
CUMU_TT[COMP_4] =
ADJ_TRANSIT_TIME[COMP_1]+ADJ_TRANSIT_TIME[COMP_2]+ADJ_TRANSIT_TIME[COMP_
3]+ADJ_TRANSIT_TIME[COMP_4]
CUMU_TT[COMP_5] =
ADJ_TRANSIT_TIME[COMP_1]+ADJ_TRANSIT_TIME[COMP_2]+ADJ_TRANSIT_TIME[COMP_
3]+ADJ_TRANSIT_TIME[COMP_4]+ADJ_TRANSIT_TIME[COMP_5]
TRANSFERS[COMP_1] = LOGN(10)/ADJ_TRANSIT_TIME[COMP_1]
TRANSFERS[COMP_2] = LOGN(10)/ADJ_TRANSIT_TIME[COMP_2]
TRANSFERS[COMP_3] = LOGN(10)/ADJ_TRANSIT_TIME[COMP_3]
TRANSFERS[COMP_4] = LOGN(10)/ADJ_TRANSIT_TIME[COMP_4]
TRANSFERS[COMP_5] = LOGN(10)/ADJ_TRANSIT_TIME[COMP_5]
USER_TT_INPUT = 1
```

United States Patent & Trademark Office

Office of Initial Patent Examination -- Scanning Division

Application deficiencies found during scanning:

☐ Page(s)__________ of ____________________ were not present for scanning.
(Document title)

☐ Page(s)__________ of ____________________ were not present for scanning.
(Document title)

☐ ***Scanned copy is best available.***

Pages number 97 to pages 121 as part of Specification are Appendicies.